# ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.3

## Аппаратные методы генерации случайных чисел с использованием микроконтроллера Arduino Nano

**В.И. Григорян, В.В. Мисюра**
Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

**Аннотация**
Настоящая работа посвящена разработке и практической реализации аппаратного генератора истинно случайных чисел на основе микроконтроллера Arduino Nano. Актуальность исследования определяется ограниченными возможностями встроенных псевдослучайных генераторов, в частности функции random среды Arduino. Целью является создание устройства, использующего физические источники энтропии для формирования случайных последовательностей. В статье описаны методы сбора и обработки энтропийных битов с двух свободных выводов. Для оценки равномерности распределения применён критерий согласия хи-квадрат с расчётом числа интервалов по формуле Стерджесса. Экспериментально подтверждено, что разработанный аппаратный генератор формирует статистически устойчивые последовательности, удовлетворяющие критерию равномерности. Новизна работы состоит в реализации статистического контроля качества генерации непосредственно на том же микроконтроллере Arduino Nano, что устраняет необходимость внешней обработки данных.

**Ключевые слова**: Arduino Nano, генератор случайных чисел, критерий хи-квадрат, равномерное распределение, формула Стерджесса, энтропия

**Для цитирования.** Григорян В.И., Мисюра В.В. Аппаратные методы генерации случайных чисел с использованием микроконтроллера Arduino Nano. *Молодой исследователь Дона.* 2026;11(4):49–53.

## Hardware Implementation of Random Number Generator Using the Arduino Nano Microcontroller

**Vladimir I. Grigoryan, Valentina V. Misyura**
Don State Technical University, Rostov-on-Don, Russian federation

**Abstract**
The article focuses on the development and practical hardware implementation of the true random number generator based on the Arduino Nano microcontroller. The relevance of the research is induced by the limited capacity of built-in pseudorandom number generators, specifically the random function in the Arduino environment. The aim of the study is to create a device that uses physical sources of entropy to generate random sequences. The article describes the methods of collecting and processing entropy bits from two free pins. To assess the uniformity of the distribution, the chi-square goodness-of-fit test was used, with the Sturges' formula used to calculate the number of intervals. The developed hardware generator had experimentally confirmed the ability to generate statistically stable sequences satisfying the uniformity criterion. The novelty of the present research lies in performing statistical control over the quality of generation using the same Arduino Nano microcontroller, which allows avoiding external data processing.

**Keywords**: Arduino Nano, random number generator, chi-square test, uniform distribution, Sturges' formula, entropy

**For Citation.** Grigoryan VI, Misyura VV. Hardware Implementation of Random Number Generator Using the Arduino Nano Microcontroller. *Young Researcher of Don.* 2026;11(4):49–53.

**Введение.** В современном мире случайные числа играют критически важную роль в самых разных областях — от компьютерного моделирования и статистического анализа до криптографии и азартных игр. Однако получение истинно случайных последовательностей на цифровых устройствах представляет собой нетривиальную задачу, поскольку компьютеры и микроконтроллеры по своей природе являются детерминированными машинами, выполняющими строго заданные алгоритмы.

*© Григорян В.И., Мисюра В.В., 2026*

Проблема заключается в том, что классические программные генераторы, доступные в стандартных библиотеках, включая функцию random() в среде Arduino, формируют лишь псевдослучайные числа [1]. Такие генераторы, основанные на линейных конгруэнтных алгоритмах, создают последовательности, которые внешне выглядят случайными, однако в действительности являются полностью детерминированными и предсказуемыми при известном начальном значении [2].

Платформа Arduino, благодаря своей доступности и наличию аналоговых входов, открывает возможности для создания аппаратных генераторов случайных чисел (True Random Number Generators, TRNG) [3]. Основная идея заключается в использовании физических процессов — таких как электромагнитные наводки, температурные изменения и собственные шумы — в качестве источника непредсказуемости. Аналоговые пины микроконтроллера в «плавающем» состоянии способны улавливать этот хаотический сигнал, который после оцифровки может служить источником энтропии.

Однако простого считывания значений с неподключенного пина недостаточно. Сразу возникает вопрос: как убедиться, что получаемые числа действительно являются случайными величинами? Здесь на помощь приходят методы математической статистики, а именно критерий хи-квадрат, который позволяет оценить равномерность распределения сгенерированных чисел.

Генераторы случайных чисел делятся на два типа: псевдослучайные (PRNG) и истинно случайные (TRNG).

Псевдослучайные генераторы — это детерминированные алгоритмы, которые создают последовательность чисел, похожую на случайную. Примером является линейный конгруэнтный генератор, описываемый формулой:

$$x_{n+1} = \left(a \cdot x_n + c\right)\left(mod m\right), \tag{1}$$

где $x_n$ — текущее число последовательности; $x_{n+1}$ — следующее число, полученное после вычисления; $a$ — множитель; $c$ — приращение; $m$ — модуль.

В Arduino функция random() реализована именно по такому принципу. Последовательность у PRNG полностью определяется начальным значением (seed) и повторяется после исчерпания периода генерации.

Истинно случайные генераторы (TRNG) используют физическую энтропию: тепловой шум резисторов, шум полупроводниковых переходов, случайные отклонения напряжения или радиоактивный распад.

Для проверки равномерности распределения последовательности применяется критерий согласия $\chi^2$ [4]. Область значений разбивается на $k$ интервалов. Для каждого интервала определяется наблюдаемая частота $O_i$ и ожидаемая частота $E_i = N/k$ ($N$ — объём выборки). Статистика критерия:

$$\chi^2 = \Sigma\left[\frac{\left(O_i - E_i\right)^2}{E_i}\right], \tag{2}$$

Полученное значение сравнивается с критическим значением $\chi^2_{крит}$, определяемым по таблицам распределения $\chi^2$ для заданного уровня значимости α (обычно 0,05) и числа степеней свободы $df = k - 1$. Если $\chi^2 < \chi^2_{крит}$ гипотеза о равномерном распределении принимается.

Важным является выбор количества интервалов $k$. Слишком малое число интервалов приводит к потере информации о характере распределения, слишком большое — к появлению интервалов с недостаточным наполнением, что снижает достоверность критерия. Оптимальное значение определяется по формуле Стерджесса [5].

В качестве аппаратной основы для генератора истинно случайных чисел была выбрана платформа Arduino Nano, обладающая необходимыми аналоговыми входами и достаточной вычислительной мощностью для выполнения статистических расчетов в реальном времени. Ключевым элементом разработанного устройства является использование аналоговых пинов (A0, A1) в качестве источников аппаратной энтропии. Эти пины остаются неподключенными, что позволяет им работать в режиме «плавающего входа» и улавливать электромагнитные наводки, тепловой шум полупроводниковых элементов и собственные случайные отклонения аналогово-цифрового преобразователя. Для визуализации результатов используется OLED-дисплей с диагональю 0.96 дюйма и разрешением 128×64 пикселей, подключенный по интерфейсу I2C (пины SDA — A4, SCL —A5) с питанием VCC — 5V и общей землей GND. На рисунке 1 представлено разработанное устройство.

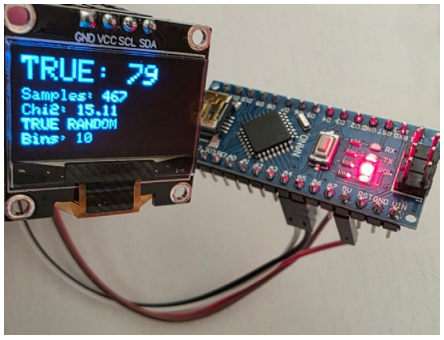


Рис. 1. Устройство для генерации случайных величин

Программная реализация аппаратного сбора энтропии основана на функции collectHardwareEntropyBit, которая поочередно считывает значения с аналоговых пинов и выделяет их младшие биты, содержащие наибольшую долю хаотической составляющей. Выбор именно младшего бита обусловлен тем, что он наиболее чувствителен к незначительным флуктуациям напряжения и поэтому содержит наибольшую случайную составляющую по сравнению со старшими битами, которые изменяются значительно реже и обладают большей устойчивостью. Для повышения качества случайности выполняется операция XOR между этими битами, что позволяет нивелировать возможные систематические отклонения отдельно взятого пина. Даже если один из входов временно демонстрирует слабую вариативность, второй источник способен компенсировать этот недостаток.

Дополнительно в алгоритме предусмотрен механизм контроля повторяющихся измерений. Если новые значения полностью совпадают с предыдущими показаниями и не содержат изменений, такой цикл не используется для генерации случайного бита. Бит считается корректно сформированным только при наличии хотя бы одного изменившегося входного значения. Это предотвращает эффект «залипания», возникающий при кратковременной стабилизации сигнала или недостаточном уровне внешних шумов.

Формирование итогового случайного числа происходит путем накопления шестнадцати таких энтропийных битов с последующим сложением по модулю с текущим значением микросекундного таймера, что добавляет временную неопределенность и исключает возможность предсказания последовательности даже при частичной компрометации источника энтропии. На рис. 2 представлена реализованная функция сбора энтропии.

```
bool collectHardwareEntropyBit() {
  static int lastValue1 = 0, lastValue2 = 0;

  int value1 = analogRead(A0);
  int value2 = analogRead(A1);

  int bit1 = value1 & 0x01;
  int bit2 = value2 & 0x01;

  int entropyBit = bit1 ^ bit2;

  bool valuesChanged = (value1 != lastValue1) || (value2 != lastValue2);
  lastValue1 = value1;
  lastValue2 = value2;

  return entropyBit && valuesChanged;
}
```

Рис. 2. Функция сбора энтропии

Для количественной проверки качества формируемой последовательности в разработанном устройстве реализован встроенный модуль статистического контроля, основанный на автоматическом вычислении критерия согласия Пирсона $\chi^2$. Использование данного метода позволяет в процессе работы генератора оценивать, насколько распределение получаемых чисел соответствует теоретически ожидаемому равномерному закону.

Перед началом вычислений весь диапазон возможных значений разбивается на несколько интервалов одинаковой ширины. Число интервалов определяется автоматически по формуле Стерджесса, которая учитывает объём анализируемой выборки и позволяет подобрать рациональное количество групп для последующей обработки данных. Так, например, для установленного максимального объёма выборки $N = 1000$ получаем значение $k \approx 11$, поэтому анализ выполняется по одиннадцати интервалам.

В ходе генерации каждое новое число относится к соответствующему интервалу, после чего счётчик данного диапазона увеличивается на единицу. Для хранения этих данных используется специальный массив частот, отражающий реальное распределение накопленной выборки. После регистрации каждой очередной сотни значений выполняется промежуточная статистическая проверка, что позволяет отслеживать изменение качества последовательности в реальном времени. Расчёт статистики $\chi^2$ производится по формуле (2).

Если используется 11 интервалов, число степеней свободы для критерия составляет: $df = k - 1 = 10$. Для уровня значимости $\alpha = 0{,}05$ табличное критическое значение равно 18,31. Если рассчитанная статистика $\chi^2$ не превышает данный порог, отклонения от равномерного распределения считаются статистически несущественными, а гипотеза о равномерности выборки не отвергается. В противном случае фиксируется ухудшение распределения, что может свидетельствовать о наличии систематического смещения либо недостаточной энтропии источника случайности.

Реализация автоматического пересчёта критерия непосредственно в микроконтроллере позволяет отказаться от внешней обработки данных на компьютере и обеспечивает автономный контроль качества генератора в процессе его функционирования. Это особенно важно для экспериментальных и учебных устройств, где требуется наглядно наблюдать изменение статистических характеристик последовательности по мере накопления выборки. На рис. 3 представлена промежуточная визуализация результатов генерации случайных чисел на OLED-дисплее.

Рис. 3. Процесс генерации случайных величин

Проведённые испытания разработанного устройства позволили установить характерные особенности работы генератора при использовании различных методов формирования случайной последовательности. На первом этапе была исследована работа встроенной функции random(), применяемой в стандартной среде Arduino. При её использовании с регулярной переинициализацией начального значения наблюдалось постепенное увеличение статистики $\chi^2$ по мере роста объёма выборки. Подобное поведение указывает на накопление отклонений от равномерного распределения и свидетельствует о том, что получаемая последовательность обладает ограниченными статистическими свойствами, характерными для псевдослучайных алгоритмов.

Рост значения критерия можно объяснить тем, что программный генератор формирует числа по заранее заданному детерминированному алгоритму. При повторных запусках и неудачном выборе начального состояния возможны повторяющиеся структуры и корреляции между соседними значениями, которые становятся заметны при анализе больших выборок.

На следующем этапе был реализован переход к аппаратному способу генерации, основанному на извлечении энтропии из шумовых процессов аналоговых входов микроконтроллера. Использование реального физического источника случайности, а также последующая обработка битов позволили существенно улучшить характеристики последовательности. В ходе экспериментов значения $\chi^2$ перестали демонстрировать устойчивый рост и колебались в пределах диапазона, близкого к теоретически ожидаемому для равномерного распределения.

Дополнительным преимуществом разработанной системы стала встроенная визуализация результатов на OLED-дисплее, обеспечивающая оперативный контроль работы устройства без подключения к внешнему компьютеру. В процессе генерации на экран выводятся текущее сформированное число, количество уже накопленных измерений, текущее значение статистики $\chi^2$, а также текстовый вывод о соответствии либо несоответствии выборки равномерному закону распределения. Подобный способ отображения информации делает устройство удобным не только для практического применения, но и для демонстрационных и учебных целей, поскольку пользователь может наблюдать изменение статистических параметров в реальном времени и сравнивать эффективность различных методов генерации случайных чисел. На рис. 4 представлен результат статистической проверки после генерации выборки объёмом 1000 случайных чисел.

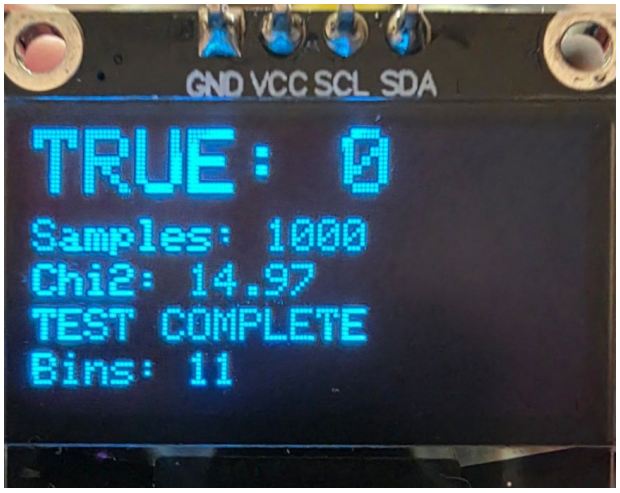


Рис. 4. Результат статистической проверки после генерации выборки объёмом 1000 случайных чисел

https://mid-journal.ru

Таким образом, в ходе проведённой работы был спроектирован и изготовлен действующий аппаратный генератор истинно случайных чисел на основе микроконтроллера Arduino Nano. В качестве источника случайности использованы физические шумовые процессы, возникающие на аналоговых входах, находящихся в неподключённом состоянии.

Разработанное устройство обеспечивает формирование случайных чисел в заданном диапазоне, отображение текущих результатов на OLED-дисплее, а также автоматический контроль статистических характеристик последовательности непосредственно в процессе работы. Для оценки качества распределения применён критерий согласия $\chi^2$, позволяющий оперативно определять степень соответствия выборки равномерному закону.

Проведённые эксперименты показали, что использование аппаратного источника энтропии с последующим накоплением и обработкой случайных битов обеспечивает более устойчивые статистические свойства по сравнению со встроенными псевдослучайными средствами платформы Arduino. Значения контрольного критерия сохранялись в допустимых пределах, что подтверждает работоспособность предложенного подхода.

Полученные результаты позволяют рассматривать созданную систему как доступное и эффективное решение для учебных лабораторных работ, демонстрационных стендов, игровых устройств и иных проектов, где требуется формирование случайных числовых последовательностей без использования внешних генераторов случайности.

**Заключение.** В результате выполненной работы был разработан и практически реализован аппаратный генератор истинно случайных чисел на базе микроконтроллера Arduino Nano, использующий физические источники энтропии — шум аналоговых входов в неподключённом состоянии. Созданное устройство обеспечивает генерацию случайных чисел в заданном диапазоне с возможностью наглядной визуализации результатов на OLED-дисплее и проведения автоматической статистической проверки качества распределения с помощью критерия хи-квадрат. Экспериментально подтверждено, что применение аппаратного сбора энтропии с последующим накоплением случайных битов позволяет получать последовательности, удовлетворяющие критериям равномерности распределения, в отличие от использования встроенных псевдослучайных генераторов. Разработанная система может найти применение в учебных целях, при создании игровых устройств, а также в любых проектах, требующих непредсказуемых последовательностей.

**Список литературы**

1. Иванов М.А., Чугунков И.В. *Теория, применение и оценка качества генераторов псевдослучайных последователеьностей*. Москва: КУДИЦ-ОБРАЗ; 2003. 240 с.

2. Подорожный И.В. Обзор аппаратных генераторов случайных чисел. *Молодой ученый.* 2016;(1(105)):190–194.

3. *Arduino.ru.* URL: http://arduino.ru (дата обращения 12.11.2025).

4. Григорьев А.Ю. Методы тестирования генераторов случайных и псевдослучайных последовательностей. *Ученые записки УлГУ. Сер. Математика и информационные технологии.* 2017;(1):22–28.

5. Р 50.1.033-2001. *Рекомендации по стандартизации. Прикладная статистика. Правила проверки согласия опытного распределения с теоретическим.* Ч. 1. Критерии типа хи-квадрат. М.: Изд-во стандартов. 2002, 87 с.

***Об авторах:***

**Владимир Иванович Григорян,** магистрант кафедры «Математика и информатика» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), forest_naight@mail.ru

**Валентина Владимировна Мисюра,** кандидат физико-математических наук, доцент кафедры «Математика и информатика» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), vvmisyura2011@gmail.com

***Конфликт интересов:*** **авторы заявляют об отсутствии конфликта интересов.**

***Все авторы прочитали и одобрили окончательный вариант рукописи.***

***About the Authors:***

**Vladimir I. Grigoryan,** Master's Degree Student, Department of Mathematics and Computer Science, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), forest_naight@mail.ru

**Valentina V. Misyura,** Cand.Sci. (Physics and Mathematics), Associate Professor of the Mathematics and Computer Science Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation) vvmisyura2011@gmail.com

***Conflict of Interest Statement:*** **the authors declare no conflict of interest.**

***All authors have read and approved the final manuscript.***